

Práctico II

Alumno: Martín La Loggia

1. ¿Algoritmo y Programa son lo mismo? Defina y Explique.

No, un algoritmo es una secuencia de pasos o instrucciones escritos en código o en lenguaje natural, que se crean para describir de forma sistemática la ejecución de una determinada tarea. En cambio, un programa es una secuencia lógica de instrucciones escritas en lenguajes de programación. Los programas se pueden componer de uno o varios algoritmos.

2. ¿Qué tipos de datos definidos por el lenguaje de Python conoce? Enumere. ¿Qué tamaño permiten, cuál es por ejemplo el mayor número entero que se puede ingresar? Investigue, Google.

Cadena: nos permite ingresar textos, palabras, letras. El tamaño de estas depende de la computadora, pero en general no pueden sobrepasar las 3GB y 2GB de memoria.

Entero: todos los números naturales. El mayor número entero es $2^{31} - 1$.

Flotante: son todos los números racionales. El mayor número flotante es $2^{31} - 1$.

Booleano: representan valores de lógica binaria, estos pueden ser de dos tipos, True o False. Es decir, verdadero o falso.(bool)

Lista: es una estructura de datos formada por una secuencia ordenada de objetos. Se accede a los elementos de una lista por su índice o también por su valor, mediante distintos métodos.

Tupla: es una colección de datos cuyo orden es inalterable. Es decir, son una lista pero que no se puede modificar, sino que tan solo leer sus elementos.

Diccionario: nos permiten almacenar una serie de mapeos entre dos conjuntos de elementos, llamados keys and values (Claves y Valores).

```
cadena = str('Hola Mundo') # Cadena de texto
entero = int(1) # Entero
flotante = float(1.5) # Flotante
booleano = bool(True) # Booleano
lista = list([1,2,3]) # Lista
tupla = tuple((1,2,3)) # Tupla
diccionario = dict(a=1,b=2,c=3) # Diccionario
```

3. **Investigue en Internet sobre los tipos de datos de C, y la especificación de cada uno. Compare con los de Python. ¿Encuentra diferencias? ¿En cuáles? Describa.**

Char: almacena caracteres o cadenas de texto.

Int: todos los números naturales.

Short int: números naturales pequeños.

Long int: números naturales grandes.

Float: números decimales.

Double: números decimales más largos.

Bool: representan valores de lógica binaria, estos pueden ser de dos tipos, true o false. Es decir, verdadero o falso.

Arreglo: son estructuras de datos consistentes en un conjunto de datos del mismo tipo. Tienen un tamaño que es la cantidad de objetos del mismo tipo que pueden almacenar.

Las principales diferencias del lenguaje C con Python, las podemos ver en los arreglos, ya que en C es necesario especificar el tipo de dato que contiene un arreglo y la cantidad de objetos del mismo. También podemos ver que en C es necesario indicar de que tipo es la variable a definir. Podemos ver que existen distintos tipos de datos para los números enteros y flotantes de acuerdo al tamaño y longitud de estos.

```
// Tipos de datos en C

char caracter = 'Hola Mundo'; // caracter o cadenas de texto
int entero = 10; // enteros
short int entero_corto = 10; // enteros cortos
long long entero_largo = 10000000000; // enteros largos
float real = 10.5; // reales o flotantes
double doble = 10.5; // dobles o flotantes dobles
bool booleano = true; // booleano
int arreglo[10] = {1,2,3,4,5,6,7,8,9,10}; // arreglo de enteros
```

4. Determine para cada punto, si es correcto o no. Justifique en cada caso, su respuesta.

a) $n = \text{int}(\text{"3.41"})$

Incorrecto, ya que 3.41 es un número por lo tanto no se deben poner las comillas. A parte tampoco es un entero, ya que es un número decimal. Lo correcto sería: $\text{float}(3.41)$ o bien $\text{str}(\text{"3.41"})$ si se lo quiere definir como un texto.

b) $p = \text{str}(\text{"hola"})$ Correcto

c) $n = \text{"23"} + \text{str}(12)$

Correcto, si lo que se busca es una concatenación, es decir, 2312. Si se lo quería definir como una suma de dos números sería: $n = 23 + 12$ o bien $n = \text{int}(23+12)$

d) $k = \text{str}(10) + \text{str}(533)$ Correcto, si lo que se busca es una concatenación de dos textos: 10533.

Ing. Carolina Zapata

Prof. Lucas Candia

5. **Coloque uno o dos pares de paréntesis en las siguientes expresiones aritméticas, para que su valor final resulte ser el que se da como Respuesta correcta para cada punto.**

- a) $3 * (23 - 7) + 186$ *Rta: 234*
b) $(185 + 53)/2 + 5$ *Rta: 124*
c) $(314 + 21 - 117)/2$ *Rta: 109*
d) $48 + 2 * (5 + 58/2) + 68$ *Rta: 184*

6. **Marque las afirmaciones que considere verdaderas.**

Un Algoritmo no puede ser implementado en Python y Java al mismo tiempo.

Los Algoritmos son independientes del lenguaje, lo que significa que un algoritmo puede ser implementado usando lenguajes diferentes.

Si la descripción de un algoritmo es ambiguo, la computadora podrá analizarlo y resolver por sí misma lo que necesita hacer.

Los algoritmos deben ser descriptos con precisión y sin ambigüedades.

7. **Cuál de las siguientes es una definición apropiada de Algoritmo:**

- a) Una nota que describe cómo llegar desde un punto de la ciudad a otro.
b) Una secuencia de pasos que se deben realizar para obtener un cierto resultado útil, en un tiempo finito.
c) Un hechizo que se debe pronunciar para lograr cierto resultado.
d) Una secuencia de acciones que se deben realizar para cocinar una torta.

8. **¿Qué tipo de dato usarías para representar cada uno de los siguientes datos? Intenta buscar más de una respuesta para cada punto, analizando en qué diferentes casos podrías necesitar representarlo con un tipo u otro para ese dato.**

- a) Número de días desde el inicio del año. **Entero o cadena.**
b) Tiempo transcurrido desde el inicio del año hasta hoy, en días. **Flotante**
c) El número serial de un celular o aparato electrónico. **Entero**
d) La edad de tu mascota. **Entero**
e) La cantidad actual total de habitantes de una ciudad. **Entero**
f) El promedio temporal de habitantes en una ciudad. **Flotante**
g) Un número de teléfono. **Entero o cadena**
h) Cantidad de intentos permitidos para ingresar la contraseña. **Entero**

9. **Escriba un algoritmo que determina si un número es par o impar. Utilice pseudocódigo.**

Pedir un número al usuario y guardarlo en una variable

Dividir ese número por 2 y obtener el producto

Comprobar si el producto es igual a cero mostrar un mensaje de que el número es par

Si el producto es distinto de cero mostrar un mensaje de que el número es impar

10. **Implemente el algoritmo del punto anterior en Python, pidiendo un número al usuario para determinar si es par o impar.**

```
# determinar si n es par o impar
n = int(input("Ingrese un número: "))
if n % 2 == 0:
    print("par")
else:
    print("impar")
```

11. Supongamos que nos dan 8 nombres. Queremos seleccionar de allí los nombres que empiecen con la letra M. ¿Cómo escribiríamos un algoritmo, que realice esa selección de nombres? Utilizar español primero para describir la idea de los pasos a seguir, y finalmente escribir un algoritmo en pseudocódigo, con los pasos bien precisos. ¿Encuentra algún problema conceptual que podría tener para implementarlo con los temas aprendidos hasta ahora? (NO debe implementarlo, solo debe diseñar un algoritmo).

Pedimos al usuario que ingrese 8 nombres y guardamos cada uno de forma separada como cadena o lista.

Comprobamos uno por uno si el primer carácter de cada valor del diccionario es igual a "M"

Si es igual a "M" sumamos un contador

Si no es igual a "M" el contador se mantiene igual

Al terminar de comprobar todos los nombres mostramos el número de nombres que empiezan con "M", el cual guardamos en el contador.

```
# determinar si el nombre empieza con M
nom1 = str(input("Ingrese su nombre: "))
nom2 = str(input("Ingrese su nombre: "))
nom3 = str(input("Ingrese su nombre: "))
nom4 = str(input("Ingrese su nombre: "))
count = 0

# guardamos el primer caracter de cada nombre en un diccionario
nombres={
    1: nom1[0],
    2: nom2[0],
    3: nom3[0],
    4: nom4[0]
}

# comprobamos los valores del diccionario que sean igual a "M"
for i in nombres:
    if nombres[i] == "M":
        count+=1

print("El número de nombres que empiezan con M es: ", count)
```

12. Implementar un programa que le permita al usuario ingresar su día y mes de nacimiento (no la fecha, solo el día, controlando que este entre 1 y 31) y el mes de nacimiento (como texto, enero, febrero, etc.) y determine de qué signo zodiacal es.

```
1  # pedimos al usuario su mes y día de nacimiento
2  mes = str(input('Ingresa el mes de nacimiento: ')).lower()
3  dia = int(input('Ingresa el día de nacimiento: '))
4
5  # definimos dos listas con los meses y los signos
6  meses = ['enero', 'febrero', 'marzo', 'abril', 'mayo', 'junio',
7           'julio', 'agosto', 'septiembre', 'octubre', 'noviembre',
8           'diciembre']
9
10 # comparamos el mes y el día para saber el signo
11 for i, value in enumerate(meses):
12     if (mes == meses[i] and dia >= 21 and dia <= 31):
13         print('Tu signo es:', signos[i])
14     elif (mes == meses[i] and dia >= 1 and dia <= 20):
15         print('Tu signo es:', signos[i-1])
16
```

13. Buscar 5 ejemplos en la vida real que necesiten de ordenación. Por ejemplo, los libros en una biblioteca deben mantenerse ordenados, cada vez que llega un nuevo libro.

Los productos en los supermercados necesitan estar ordenados por categorías y marcas.

La lista de alumnos de una escuela necesitan estar ordenados alfabéticamente.

Los archivos de una empresa deben mantenerse ordenados para su posterior uso.

Los turnos del médico deben estar ordenados por día y por hora.

Los clientes de una cooperativa deben estar ordenados para facilitar la búsqueda de las facturas de dichos clientes.

14. ¿Cómo ordenarías 4 números? piensa diferentes estrategias, luego intenta escribir un algoritmo basado en alguna de esas estrategias. Pásalo a pseudocódigo.

Pedimos al usuario que ingrese 4 números.

Los añadimos a una lista.

Luego ordenamos la lista de mayor a menor.

Mostramos en pantalla un mensaje con los números ordenados.

15. Intenta implementar un programa que utilice una de las estrategias anteriores para ordenar cuatro números de menor a mayor, ingresados por el usuario.

```
lista = []
count = 0
# repetimos un bucle 4 veces para pedir al usuario un número y
# añadirlo a una lista
while(count<4):
    n = int(input("Ingrese un número: "))
    lista.append(n)
    count+=1
# mostramos la lista en pantalla
print(sorted(lista))
```